APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: CALCULATING DISPLAY MODE VALUES

APPLICANT: KALPESH MEHTA, ERIC SAMSON, WEN-SHAN (VINCENT) WANG AND MICHAEL DONLON

9.

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV399289646US

January 16, 2004

Date of Deposit

CALCULATING DISPLAY MODE VALUES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation application which claims benefit of U.S. Application No. 09/579,335, filed May 25, 2000. The disclosure of the prior application is considered part of (and is incorporated by reference in) the disclosure of this application.

BACKGROUND

[0002] This invention relates to calculating display mode values.

[0003] A display streamer in a graphics processor requests display data from memory to be temporarily stored in a FIFO (first-in first-out) and continuously feeds the display data to a display engine. Any break or interruption in feeding the display data results in visual artifacts in the final output (display) on a display device, e.g., an analog cathode ray tube (CRT) monitor. Additionally, the memory is usually most efficient when providing data at a high rate while the graphics processor can usually only use data at a rate that is much lower than this high rate.

[0004] To eliminate these visual artifacts and increase efficiency, the display streamer may be programmed with a watermark value and a burst length value for each display mode

supported by the graphics processor. A display mode can be, e.g., a combination including display device resolution, color depth or pixel depth, refresh rates, and system configuration. The watermark value represents a FIFO size and falls between the minimum and maximum size of the FIFO, usually expressed in quadwords (QW) that are blocks of eight bytes each.

[0005] When the amount of data in the FIFO drops below the watermark value for the current display mode, the display streamer requests more display data from memory. A display mode's burst length value falls between the minimum and maximum amounts of display data, usually expressed in QW, that the display streamer may request from memory at a time. Analytic models may be used to predict the watermark values and burst length values for each display mode. There are over one hundred display modes.

DESCRIPTION OF DRAWINGS

[0006] FIG. 1 is a block diagram of a computer system in accordance with an embodiment of the invention.

[0007] FIG. 2 is a block diagram of a display system included in the computer system of FIG. 1.

[0008] FIG. 3 is a diagram of the display system of FIG. 2.

[0009] FIG. 4 is a flowchart of calculating and programming display mode values in accordance with an embodiment of the invention.

[0010] FIG. 5 is a graph showing display mode values.

DESCRIPTION

[0011] Referring to FIG. 1, a system 10 includes a central processing unit (CPU) 12 that computes watermark values and burst length values "on the fly" as the system 10 encounters different display modes. Different display modes result from different configurations of the system 10. A configuration can be, e.g., a particular combination of multiple displays, display resolutions, color depths, refresh rates, overlay scaling conditions, video capture conditions, and/or other system configurations. The CPU 12 programs one of the watermark values as a current watermark value and one of the burst length values as a current burst length value into a graphics controller for use in processing the graphics or video data destined for display on one or more display devices 22. The graphics controller could be included in either a graphics/memory controller (GMCH) 14 or a graphics controller (Gfx) 16 hanging on an accelerated graphics port (AGP) 18. In this embodiment, assume that the graphics controller is included in the GMCH 14. The GMCH 14 uses these values in

streaming video or graphics image data. This data can be lines of the image held in main memory, e.g., dynamic random access memory (DRAM) 20, to a display device 22, e.g., a computer monitor, a television, or a floating point display unit.

[0012] Also referring to FIG. 2, a software driver (not shown) and/or a hardware logic unit (not shown) included in the CPU 12 calculates the watermark values and burst length values using the formulas discussed below and programs a display streamer 30 in the GMCH 14 with a watermark value and a burst length value for the current display mode, the present display mode of the system 10. These values enable the display streamer 30 to more efficiently control how and when the data is fetched from any data source, including local memory 32 and/or main memory 36, e.g., DRAM or synchronous dynamic random access memory (SDRAM), and provided to a display mechanism such as a display engine 34, a device that provides the display device 22 with displayable data. Local memory 32 may be included in the GMCH 14, in the Gfx 16, or as a separate unit.

[0013] Any hardware system having a memory that can store data included in an isochronous data stream, i.e., real-time, non-display data streams, e.g., modems, LANs (local area networks), and other real-time systems with event deadlines,

can compute watermark and burst length values "on-the-fly" using the formulas below. The hardware system can use the software driver and/or the hardware logic unit to compute the watermark and burst length values and improve the efficiency of transferring the isochronous data between the memory and a destination of the isochronous data included in the hardware system.

[0014] Also referring to FIG. 3, a display FIFO 40 located between the memory controller 31 and the display engine 34 eliminates visual artifacts and smooth out delay jitters. Delay jitters manifest as flickers or breaks on the display device 22 and smoothing them out produces more pleasing video or graphics images, ones with less visual artifacts. The display FIFO 40 holds up to a certain number of quadwords (QW) of data fetched from local memory 32 or main memory 36, ready to be processed by the display engine 34 and shown on the display device 22. If the local memory 32 is a separate unit, it can connect to the memory controller 31 and use the main memory 36.

[0015] Storing QW of data in the display FIFO 40 can help increase efficiency of the data transfer between the memory and the graphics controller. The memory can provide data at one rate while the graphics controller can use data at

another, slower rate by storing data the graphics controller is not ready to use in the FIFO 40.

[0016] The maximum size of the display FIFO 40 depends on the worst case delay (maximum latency, L_{max}), the FIFO fill rate, and the FIFO drain rate. The arbitration policy in the memory controller 14 determines L_{max}. For example, the display engine 34 may be granted access to local memory 32 more frequently than other isochronous clients such as a video capture engine 42 or an overlay scaling engine 44 and more frequently than non-isochronous clients such as a two-dimensional engine 46. The value of L_{max} represents the maximum amount of time in clock cycles that the display engine 34 may have to wait before winning another arbitration event and gaining access to local memory 32 to obtain data to occupy the display FIFO 40. The speed of the SDRAM 36 determines the FIFO fill rate (φ), expressed in QW per local memory clock cycle. The FIFO drain rate (δ), expressed in QW per clock cycle, is determined by the rate at which data is consumed by the display engine 34. The display resolution and the refresh rate contribute to δ as shown below.

[0017] The display streamer 30 uses the watermark value (λ) and the burst length value (β) calculated by the driver and/or the hardware logic unit in the CPU 12 and programmed into a register included in the display streamer 30 in continuously

monitoring the level of data in the display FIFO 40 and ensuring that the display engine 34 receives a continuous flow of data. If the FIFO level falls below λ, the display streamer 30 issues a request in a burst action to local memory 32 or main memory 20, 36 for an amount of data equal to β to occupy the display FIFO 40.

[0018] The driver and/or hardware logic unit in the CPU 12 chooses λ as a value between a minimum watermark value (λ_{min}) and a maximum watermark value (λ_{max}). λ_{min} is the value which avoids FIFO underflows and delay jitter. λ_{min} is given by:

$$\lambda_{min} = L_{max} \times \delta$$

Because this formula likely returns λ_{min} as a floating point number and because computer systems operate with integers, the driver and/or hardware logic unit computes λ_{min} with a ceiling subroutine as the smallest integer value greater than the floating point value of λ_{min}. A λ_{min} at this integer value helps the display FIFO 40 avoid underflows because λ_{min} is greater than the FIFO drain during L_{max} cycles of waiting.

[0019] The amount of data in QW (β) that the display streamer 30 requests in response to detecting a data level below λ in the display FIFO 40 falls between a minimum burst

length value (β_{min}) and a maximum burst length value (β_{max}). β_{min} is given by:

$$\beta_{min} = \lambda_{min} \times \left(\frac{\varphi}{\varphi - \delta}\right)$$

As with λ_{min}, the driver and/or hardware logic unit computes β_{min} with a ceiling subroutine as the smallest integer value greater than the floating point value of β_{min}. This integer β_{min} value ensures that the display streamer 30 requests enough QW to guarantee that the level of the display FIFO 40 meets or exceeds λ_{min} at the end of the burst.

[0020] To ensure that the display FIFO 40 does not overflow, the display streamer 30 should not request more QW than a maximum burst length value (β_{max}) in a given burst. β_{max} is given by:

$$\beta_{max} = (\Phi - \lambda_{min}) \times \left(\frac{\varphi}{\varphi - \delta}\right),$$

where Φ equals the size of the display FIFO 40 in QW. Since this β_{max} formula likely returns a floating point value, the driver and/or hardware logic unit uses a floor subroutine to calculate an integer β_{max} value that is the largest integer value less than the floating point value of β_{max}.

[0021] Also to help prevent overflow, the maximum watermark level (λ_{max}) indicates the maximum amount of data that the display FIFO 40 may contain when the display streamer 30 begins a burst without overflowing the display FIFO 40 with the requested data. λ_{max} is given by:

$$\lambda_{max} = \Phi - (L_{max} \times \delta)$$

As with β_{max}, the driver and/or hardware logic unit uses a floor subroutine to calculate an integer value of λ_{max} that is the largest integer value less than the floating point value of λ_{max}.

[0022] Also referring to FIG. 4, the driver and/or hardware logic unit in the CPU 12 uses a process 50 to calculate the watermark value and the burst length value for a current display mode. The process 50 begins (52) by determining (54) any constraints of the system hardware under the current display mode from the graphics/memory controller 14, graphics controller 12, and/or the display device 22. Such constraints may include memory speed, multiple displays, overlay scaling functions, and/or video capture functions. For example, in one current display mode, the display FIFO 40 size is 48QW, local memory 32 is running at 133MHz and the worst case latency (L_{max}) for the display streamer 30 is forty cycles.

The driver and/or hardware logic unit also identifies (56) parameters of the display device 22 such as supportable resolutions, color depth, and refresh rates. In the current display mode, the display device 22 has a 1280 x 1024 resolution running at a 100Hz refresh rate in 16 bpp (bits per pixel) mode. Based on these constraints and parameters, the driver and/or hardware logic unit can calculate (58) φ, the FIFO fill rate. Assume that φ equals one in the current display mode. The driver and/or hardware logic unit may determine (54) the hardware constraints and identify (56) the display device's parameters in any order.

[0023] The driver and/or hardware logic unit then determines (60) if Φ, the size of the display FIFO 40, is large enough for a specified drain rate δ and L_{max} using the comparative formula:

$$\Phi > 2 \times L_{max} \times \delta \ ,$$

where δ equals approximately 0.357 and is given by:

$$\delta = (\text{display clock frequency}) \times \left(\frac{\text{bytes per pixel}}{\text{bytes per QW} \times \text{memory speed}} \right)$$

The display clock frequency (DCF) depends on the current display mode and can be expressed in an empirical formula as:

$$\text{DCF} = (\text{horizontal resolution}) \times (\text{vertical resolution}) \times (\text{refresh rate}) \times 1.45 \ ,$$

where 1.45 is a multiplying factor. Other methods may be used to calculate the DCF, e.g., a table-based method or a Video Electronics Standards Association generalized timing formula (VESA GTF). If Φ is not large enough, then the display FIFO 40 is too small to handle the requirements of the current display mode and the process 50 fails (62). If Φ is large enough, then the driver and/or hardware logic unit may proceed to calculate (64) the watermark value and the burst length value for the current display mode.

[0024] The driver and/or hardware logic unit calculates (64) integer values for λ_{min}, λ_{max}, β_{min}, and β_{max} as described above. In the current display mode, they respectively equal fifteen, thirty-three, twenty-four, and fifty-one. The driver and/or hardware logic unit compares (66) β_{min} and β_{max} to see if the system 10 can accommodate the current display mode. If β_{max} is less than β_{min}, then the process fails (62), and the current display mode is unsupportable. Otherwise, the driver and/or hardware logic unit compares (68) λ_{min} and λ_{max}. The driver and/or hardware logic unit may compare (66, 68) either burst length values or watermark values first. If λ_{max} is greater than λ_{min}, then the process 50 fails (62). Otherwise, the driver and/or hardware logic unit chooses (70) a watermark

value λ between λ_{min} and λ_{max} and a burst length value β between β_{min} and β_{max}.

[0025] Also referring to FIG. 5, the driver and/or hardware logic unit chooses (70) λ and β for the current display mode from within a region 80 defined by λ_{min}, λ_{max}, β_{min}, and β_{max}. All of the points within the region 80 are permissible (supportable by the system 10) λ and β pairs. The driver and/or hardware logic unit preferably chooses (70) λ and β from a point in the lower left corner of the region 80. Specifically, λ is chosen (70) as the integer value of λ_{min} and β is chosen (70) as:

$$\beta = \text{ceil}\left(\frac{\beta_{min}}{8}\right) \text{ x } 8 \text{ ,}$$

where "ceil" indicates the ceiling subroutine explained above. This equation forces β to meet or exceed β_{min} and be a multiple of eight so that the display streamer 30 can request an integer number of QW. In other embodiments, the "eights" in the above equation may equal any number, including one. Note that the region 80 shrinks for higher resolutions and refresh rates. The region 80 may not contain any permissible points indicating an unsupportable display mode. The driver and/or hardware logic unit programs (72) the chosen λ and β values into the display streamer 30 and the process 50 ends (74).

[0026] Other embodiments are within the scope of the following claims.